

Claude Code

Cheatsheet

Last updated May 2025 · code.claude.com/docs

■ CLI — Session Management

Command	Description	Example
<code>claude</code>	Start an interactive session	<code>claude</code>
<code>claude "query"</code>	Start session with initial prompt	<code>claude "explain this project"</code>
<code>claude -p "query"</code>	Print response and exit (non-interactive / SDK mode)	<code>claude -p "explain this function"</code>
<code>cat file claude -p "q"</code>	Process piped content	<code>cat logs.txt claude -p "explain"</code>
<code>claude -c</code>	Continue the most recent conversation	<code>claude -c</code>
<code>claude -r "session" "q"</code>	Resume a session by ID or name	<code>claude -r "auth-refactor" "Finish this PR"</code>
<code>claude -n "name"</code>	Name the current session	<code>claude -n "my-feature-work"</code>
<code>claude update</code>	Update to the latest version	<code>claude update</code>
<code>claude install [version]</code>	Install a specific version (stable, latest, 2.x.x)	<code>claude install stable</code>

■ CLI — Authentication

Command	Description
<code>claude auth login</code>	Sign in to your Anthropic account
<code>claude auth logout</code>	Log out from your Anthropic account
<code>claude auth status</code>	Show authentication status (--text for human-readable)
<code>claude setup-token</code>	Generate a long-lived OAuth token for CI / scripts
<code>claude auth login --sso</code>	Force SSO authentication

■■ CLI — Configuration Flags

Flag	Description
<code>--model</code>	Set model: sonnet, opus, or full name (e.g. claude-opus-4-6)
<code>--effort</code>	Effort: low, medium, high, xhigh, max
<code>--permission-mode</code>	Start in: default, acceptEdits, plan, auto, bypassPermissions
<code>--add-dir</code>	Add additional working directories for file access
<code>--append-system-prompt</code>	Append custom text to the system prompt
<code>--system-prompt "..."</code>	Replace the entire system prompt
<code>--system-prompt-file</code>	Load system prompt from a file (replaces default)

Flag	Description
--tools "..."	Restrict which tools Claude can use (e.g. Bash,Edit,Read)
--allowedTools "..."	Tools that run without a permission prompt
--disallowedTools "..."	Tools that are denied or removed from context
--mcp-config	Load MCP servers from a JSON file at launch
--plugin-dir	Load a plugin for this session only
--bare	Minimal mode — skips hooks, skills, MCP, CLAUDE.md
--bg	Start as a background agent and return immediately
--debug "cat,cat"	Enable debug mode with optional category filter
--verbose	Show full turn-by-turn output
--max-turns	Limit agentic turns (print mode only)
--max-budget-usd	Budget cap in dollars (print mode only)
--output-format	Output format: text, json, stream-json
-w / --worktree	Start in an isolated git worktree
--chrome	Enable Chrome browser integration
-v / --version	Output the version number

■ Configuration (Inside Session)

Command	Description
/config	Open Settings UI (theme, model, output style...). Alias: /settings
/model [model]	Select or change the AI model interactively
/effort [level]	Set effort level or open interactive slider. Alias: /auto
/theme	Change color theme (light, dark, daltonized, ANSI, custom)
/permissions	Manage allow/ask/deny rules for tools. Alias: /allowed-tools
/hooks	View hook configurations for tool events
/mcp	Manage MCP server connections and OAuth
/agents	Manage subagent configurations
/terminal-setup	Configure Shift+Enter and other terminal keybindings
/keybindings	Open or create your keybindings configuration file
/statusline	Configure the Claude Code status line

■ Context Management

Command	Description
/context [all]	Visualize context usage as a colored grid with tips
/compact [instructions]	Summarize conversation to free context. Example: /compact keep only the plan and diff
/clear [name]	Start a fresh conversation (previous stays in /resume). Aliases: /reset, /new

Command	Description
<code>/btw</code>	Ask a side question without adding to conversation history

■■ Information & Status

Command	Description
<code>/help</code>	Show help and all available commands
<code>/doctor</code>	Diagnose installation, settings, MCP servers, and context usage
<code>/status</code>	Show version, model, account, connectivity (works while responding)
<code>/usage</code>	Show session cost, plan limits, activity stats. Aliases: <code>/cost</code> , <code>/stats</code>
<code>/skills</code>	List available skills (press <code>t</code> to sort by token count)
<code>/memory</code>	Edit CLAUDE.md memory files and manage auto-memory
<code>/insights</code>	Analyze your Claude Code sessions for patterns and friction points
<code>/release-notes</code>	View changelog in an interactive version picker

■■ Memory & Project Setup

Command / Path	Description
<code>/init</code>	Initialize project with a CLAUDE.md guide
<code>/memory</code>	Edit CLAUDE.md files, enable/disable auto-memory
<code>/team-onboarding</code>	Generate a team onboarding guide from usage history
<code>~/.claude/CLAUDE.md</code>	Global (personal) memory — applies to all projects
<code>.claude/CLAUDE.md</code>	Project-level memory (committed to repo)
<code>.claude/settings.json</code>	Project settings
<code>.claude/settings.local.json</code>	Local (gitignored) project settings
<code>~/.claude/settings.json</code>	User-level settings

■ Session Navigation

Command	Description
<code>/resume [session]</code>	Resume a conversation by ID or name. Alias: <code>/continue</code>
<code>/branch [name]</code>	Fork the current conversation at this point. Alias: <code>/fork</code>
<code>/rename [name]</code>	Rename the current session
<code>/rewind</code>	Roll back code and conversation to a checkpoint. Aliases: <code>/checkpoint</code> , <code>/undo</code>
<code>/export [filename]</code>	Export the current conversation as plain text
<code>/clear</code>	Start a new conversation with empty context

■ Workflow Commands

Command	Description
<code>/plan [description]</code>	Enter plan mode. Pass a description to start immediately
<code>/diff</code>	Open interactive diff viewer for uncommitted changes and per-turn diffs
<code>/copy [N]</code>	Copy last assistant response to clipboard (N = Nth-latest)
<code>/goal [condition]</code>	Set a persistent goal; Claude works across turns until met
<code>/background [prompt]</code>	Detach session to run as a background agent. Alias: <code>/bg</code>

■ Parallel Work & Agents

Command	Description
<code>/agents</code>	Open agent manager to monitor and dispatch parallel sessions
<code>/tasks</code>	List and manage background tasks. Alias: <code>/bashes</code>
<code>/batch</code>	[Skill] Break large change into parallel units in isolated worktrees
<code>/background</code>	Detach current session to run in the background. Alias: <code>/bg</code>
<code>claude agents</code>	Open agent view from the CLI
<code>claude attach</code>	Attach to a background session in this terminal
<code>claude stop</code>	Stop a background session
<code>claude logs</code>	Print recent output from a background session
<code>claude respawn</code>	Restart a background session with its conversation intact
<code>claude rm</code>	Remove a background session from the list

■ Code Quality — Bundled Skills

Command	Description
<code>/review [PR]</code>	Review a pull request locally in the current session
<code>/security-review</code>	Analyze pending changes for security vulnerabilities
<code>/simplify [focus]</code>	[Skill] Review recently changed files for quality issues, then fix them
<code>/ultrareview [PR]</code>	Deep multi-agent cloud-based code review
<code>/debug [description]</code>	[Skill] Enable debug logging and troubleshoot session issues
<code>/fewer-permission-prompts</code>	[Skill] Scan transcripts and add allowlist rules to reduce prompts

■■ Debugging & Diagnostics

Command	Description
<code>/doctor</code>	Run automated check of installation, settings, MCP, and context
<code>claude doctor</code>	Same as <code>/doctor</code> , but from the shell (use if claude won't start)
<code>/debug [description]</code>	Enable debug logging mid-session
<code>/heapdump</code>	Write JS heap snapshot to <code>~/Desktop</code> for memory diagnostics

Command	Description
<code>/feedback [report]</code>	Submit feedback or bug report to Anthropic. Alias: <code>/bug</code>
<code>claude --debug</code>	Start with debug output (optional category filter e.g. <code>api,mcp</code>)
<code>claude --debug-file</code>	Write debug logs to a specific file path

■ MCP & Plugins

Command	Description
<code>/mcp</code>	Manage MCP server connections and OAuth inside a session
<code>claude mcp</code>	Configure MCP servers from the CLI
<code>/plugin</code>	Manage Claude Code plugins
<code>claude plugin install</code>	Install a plugin from the CLI
<code>/reload-plugins</code>	Reload all active plugins without restarting
<code>--mcp-config</code>	Load MCP servers from a JSON file at launch
<code>--strict-mcp-config</code>	Use only MCP servers from <code>--mcp-config</code> ; ignore all others

■ Remote & Web

Command	Description
<code>/remote-control</code>	Make this session available for remote control from <code>claude.ai</code> . Alias: <code>/rc</code>
<code>claude remote-control</code>	Start a Remote Control server (CLI)
<code>claude --remote "task"</code>	Create a new web session on <code>claude.ai</code>
<code>/teleport</code>	Pull a Claude Code web session into this terminal. Alias: <code>/tp</code>
<code>/autofix-pr [prompt]</code>	Spawn a web session that watches the current PR and pushes fixes
<code>/desktop</code>	Continue this session in the Claude Code Desktop app

■ Custom Skills — Key Locations & Frontmatter

Path / Field	Description
<code>~/.claude/skills//SKILL.md</code>	Personal skill — available across all projects
<code>.claude/skills//SKILL.md</code>	Project skill — this project only
<code>description:</code>	Frontmatter: Claude uses this to auto-invoke the skill
<code>disable-model-invocation: true</code>	Frontmatter: only you can invoke this skill (no auto-trigger)
<code>user-invocable: false</code>	Frontmatter: only Claude can invoke (hidden from <code>/</code> menu)
<code>allowed-tools: Bash(git *)</code>	Frontmatter: pre-approve tools for this skill
<code>context: fork</code>	Frontmatter: run skill in an isolated subagent
<code>effort: high</code>	Frontmatter: override effort level for this skill

■ Keyboard Shortcuts (Interactive Mode)

Shortcut	Action
Ctrl+C	Cancel current operation
Shift+Tab	Cycle permission modes: default → plan → auto → bypassPermissions
Ctrl+R	Search prompt history
↑ / ↓	Navigate prompt history
Tab	Autocomplete skill / command name

■ Quick Tips

- Type **/** in a session to see all commands with live filtering.
- Use **Shift+Tab** to cycle permission modes without restarting.
- **/compact** with instructions (e.g. */compact keep only the plan*) lets you surgically preserve context.
- **CLAUDE.md** is read at every session start — keep it focused and under 500 lines.
- **claude -p** (print mode) is ideal for scripting and CI pipelines.
- Background sessions (**--bg**) let you run long tasks without tying up a terminal.
- Set **disable-model-invocation: true** in a skill's frontmatter to require manual invocation.
- **/batch** is the power move for large cross-codebase changes — it parallelizes work across worktrees.